CLAIMS

We Claim

1. A method of generating identifier data for persistently identifying a user interface element of interest in a graphical user interface of a source computer program, the method comprising:

receiving data indicative of the user interface element of interest from a first software component; and

in response to receiving the data indicative of the user interface element of interest, generating an element path identifier of the user interface element of interest for persistently identifying the user interface element of interest and returning at least the element path identifier to the first software component.

2. The method of claim 1, wherein generating the element path identifier is implemented by a second software component communicative with the source computer program.

3. The method of claim 1, wherein generating an element path identifier comprises:

using a hierarchical tree structure representation of the graphical user interface to locate a leaf node related to the user interface element of interest;

storing exposed identifier information of the user interface element of interest in an element path identifier data structure; and

proceeding up the hierarchy of the tree structure representation to store the exposed identifier information of selected parent nodes of the user interface element of interest in the element path identifier data structure.

4. The method of claim 3, further comprising converting the element path identifier to a string type data structure.

5. The method of claim 3, further comprising receiving data indicative of a designated element path root node of the tree structure representation, wherein the step of proceeding up the hierarchy of the tree structure representation to store the exposed identifier information of selected parent nodes is continued only until the element path root node is reached.

6. The method of claim 3, wherein the exposed identifier information is a local alpha numeric identifier not guaranteed to be unique, a class name, a module name associated with an application program or a sibling order.

7. The method of claim 3, further comprising:

determining that at least one of the selected parent nodes is a root node of a strongly named branch portion of the tree structure representation, wherein the strongly named branch portion has at least one user interface element within a scope inside of which the at least one user interface element is guaranteed to be uniquely identifiable by a named branch element identifier; and

storing in the element path identifier data structure, the named branch element identifier for the at least one user interface element within the scope of the named branch portion.

8. The method of claim 7, further comprising:

storing in the element path identifier a strong name associated with the strongly named branch along with the named branch element identifier.

9. The method of claim 7, wherein the named branch element identifier stored in the element path identifier data structure is for the user interface element of

interest and no identifying information related to the parent elements within the scope are stored in the element path identifier data structure.

10. A method of searching for a target element in a graphical user interface of a target computer program, the method comprising:

receiving an element path identifier of the target element from a first software component; and

using a hierarchical tree structure representation of the graphical user interface to locate the target element in the graphical user interface hierarchy based on its element path identifier.

11. The method of claim 10, wherein locating the target element in the graphical user interface hierarchy is implemented by a second software component communicative with the target computer program.

12. The method of claim 10, wherein the element path identifier is a string type data structure.

13. The method of claim 10, wherein the element path identifier comprises identifier information of elements of an element path related to the target element.

14. The method of claim 10, wherein the element path identifier comprises class names of component elements of an element path of the target element.

15. The method of claim 10, wherein the element path identifier comprises a module name of an application program related to component elements of an element path of the target element.

16. The method of claim 10, wherein the element path identifier comprises sibling order data.

17. The method of claim 10, wherein locating the target element comprises:

locating an element path root node within the hierarchical tree structure representation of the graphical user interface;

comparing appropriate fields of a data structure of the element path identifier to identifier data exposed by the element path root node to determine a match; and

upon determining the match, proceeding to nodes in lower levels of the hierarchical tree structure representation to compare the identifier information respectively exposed by the nodes in the lower levels of the hierarchical tree structure with appropriate fields of the element path identifier to determine nodes that match until a node of the target element is found.

18. The method of claim 17, wherein the element path identifier comprises data indicative of the element path root node.

19. The method of claim 17, further comprising:

determining that at least one node in the hierarchical tree structure representation is a root node of a strongly named branch portion of the tree structure representation having a scope, wherein elements within the scope are guaranteed to be uniquely identifiable by corresponding named branch element identifiers; and

using the named branch element identifiers for determining nodes that match.

20. The method of claim 19, wherein the element path identifier comprises named branch element identifiers of elements within the scope of the strongly named branch portion.

21. At least one computer-readable medium having stored thereon computer-executable instructions related to a function responsive to a function call from a first software component, the function comprising:

an input parameter representing a user interface element of interest in a graphical user interface of a source computer program;

an output parameter representing an element path identifier for persistent identification of the element of interest; and

executable software for receiving the input parameter representing an user interface element of interest and in response, generating the output parameter representing an element path identifier of the user interface element of interest.

22. The computer-readable medium of claim 21, wherein the element path identifier is a string type data structure.

23. The computer-readable medium of claim 21, wherein the element path identifier comprises exposed identifier information of component elements of an element path related to the element of interest.

24. The computer-readable medium of claim 21, wherein the element path identifier comprises class names of component elements of an element path of the element of interest.

25. The computer-readable medium of claim 21, wherein the element path identifier comprises a module name of an application program related to component elements of an element path of the element of interest.

26. The computer-readable medium of claim 21, wherein the element path identifier comprises sibling order data.

27. The computer-readable medium of claim 21, wherein generating the output parameter representing an element path identifier comprises determining that at least one node in a hierarchical tree structure representation of the graphical user interface of the source computer program is a root node of a strongly named branch portion of the tree structure representation, wherein the strongly named branch portion has at least one user interface element within a scope inside of which the at least one user interface element is guaranteed to be uniquely identifiable by a named branch element identifier, and the element path identifier comprises the named branch element identifier for the at least one user interface element within the scope of the named branch portion.

28. At least one computer-readable medium having stored thereon computer-executable instructions related to a function responsive to a function call from a first software component, the function comprising:

an input parameter representing an element path identifier of a target user interface element in a graphical user interface of a target computer program;

an output parameter representing a location of the target user interface element within a hierarchical tree structure representation of the graphical user interface; and

executable software for receiving the element path identifier of the target user interface element and determining the output parameter representing the location of the target user interface element.

29. The computer-readable medium of claim 28, wherein the element path identifier is a string type data structure.

30. The computer-readable medium of claim 28, wherein the element path identifier comprises exposed identifier information of component elements of an element path related to the element of interest.

31. The computer-readable medium of claim 28, wherein the element path identifier comprises class names of component elements of an element path of the element of interest.

32. The computer-readable medium of claim 28, wherein the element path identifier comprises a module name of an application program related to component elements of an element path of the element of interest.

33. The computer-readable medium of claim 28, wherein the element path identifier comprises sibling order data.

34. The computer-readable medium of claim 28, wherein hierarchical tree structure representation of the graphical user interface of the target computer program comprises at least one node that is root node of a strongly named branch portion of the tree structure representation, wherein the strongly named branch portion has at least one user interface element node within a scope inside of which the at least one user interface element node is guaranteed to be uniquely identifiable by a named branch element identifier and the input parameter representing the element path identifier comprises the named branch element identifier of the at least one user interface element guaranteed to be uniquely identifiable.

35. In a computer system running a computer program with a graphical user interface, a system for generating element path identifiers of elements of the graphical

user interface and later searching for the elements of the graphical user interface using the element path identifiers, the system comprising:

an API module comprising a first set of APIs related to passing function calls for generating the element path identifiers and a second set of APIs related to passing function calls for searching for the elements of the graphical user interface using the element path identifiers; and

an element path engine responsive to the function calls for generating the element path identifiers and to the function calls for searching for the elements of the graphical user interface using the element path identifiers.